USF UNIVERSITY OF SAN FRANCISCO

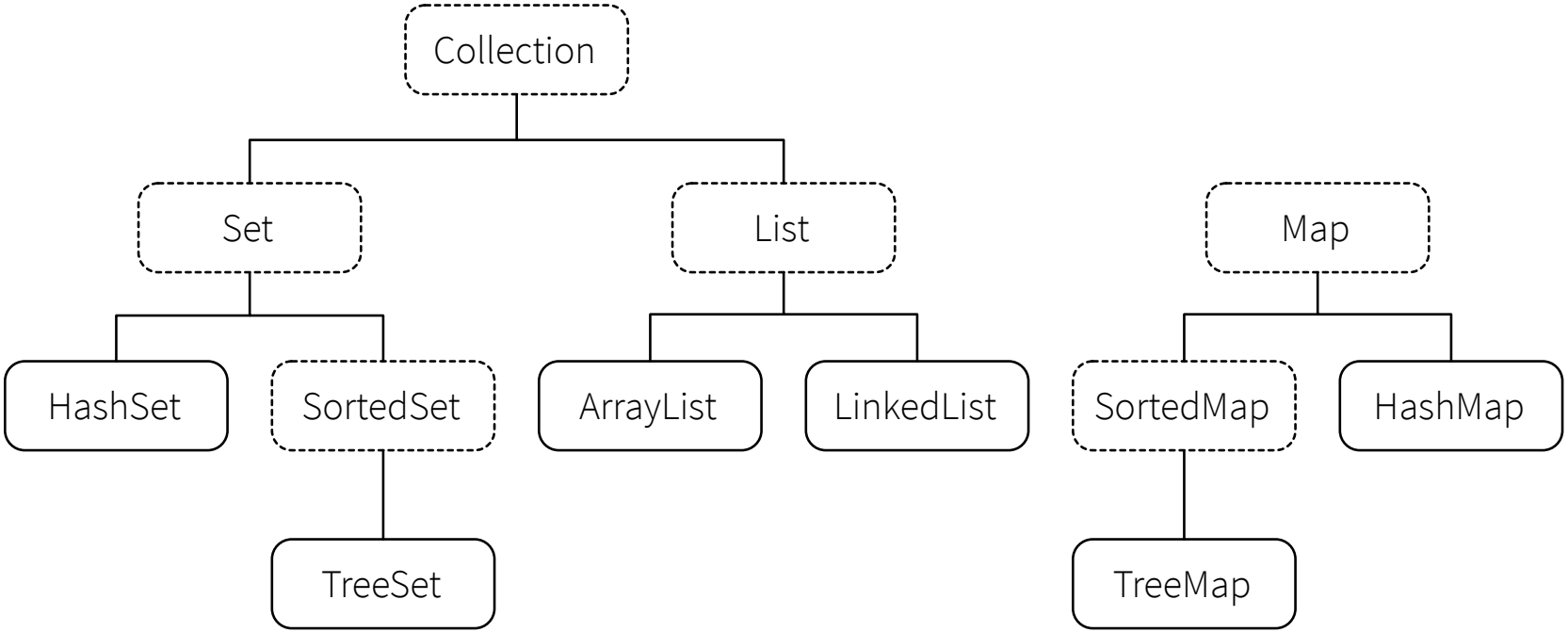
CHANGE THE WORLD FROM HERE

Inheritance

Collections Revisited

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science

Collection Framework*



*Abbreviated Framework

ArrayList Revisited

java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractList<E>
      java.util.ArrayList<E>

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

More Complex Than Previously Advertised

```
public class ArrayList<E>
extends AbstractList<E>
implements List<E>, RandomAccess, Cloneable, Serializable
```

Resizable-array implementation of the `List` interface. Implements all optional list operations, and permits all elements, including `null`. In addition to implementing the `List` interface, this class provides methods to manipulate the size of the array that is used internally to store the list. (This class is roughly equivalent to `Vector`, except that it is unsynchronized.)

The `size`, `isEmpty`, `get`, `set`, `iterator`, and `listIterator` operations run in constant time. The `add` operation runs in *amortized constant time*, that is, adding n elements requires O(n) time. All of the other operations run in linear time (roughly speaking). The constant factor is low compared to that for the `LinkedList` implementation.

Each `ArrayList` instance has a *capacity*. The capacity is the size of the array used to store the elements in the list. It is always at least as large as the list size. As elements are added to an ArrayList, its capacity grows automatically. The details of the growth policy are not specified beyond the fact that adding an element has constant amortized time

https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html

ArrayList Revisited

- Interface Collection extends the Iterable interface
- Interface List extends Collection
- Class AbstractCollection implements Collection
- Class AbstractList extends AbstractCollection and implements List (and hence Collection)
- Class ArrayList extends AbstractList and implements other interfaces on top of List and Collection

https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html

Collection Interface

- Root of collection hierarchy is an interface!
- Includes methods such as add(), clear(), contains(), remove(), size(), toArray()
- Method iterator() inherited from Iterable
  - Allows any collection to be used in for-each loops

https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html

List Interface

- Extends Collection interface
  - And thus also inherits from Iterable
- Adds positional methods to get, insert, modify, or remove elements by position
- Adds ability to create a sublist

https://docs.oracle.com/javase/8/docs/api/java/util/List.html

AbstractCollection Class

- An abstract class that implements Collection
- Optional methods all throw an unsupported operation exception (discussed later)
- Provides skeleton implementations of other methods except iterator() and size()

https://docs.oracle.com/javase/8/docs/api/java/util/AbstractCollection.html

AbstractList Class

- An abstract class that extends AbstractCollection and implements List (and hence Collection)
- Optional methods still throw exceptions
- Provides iterator implementations for any list
- Provides skeletal implementations for all except get() and size() from AbstractCollection

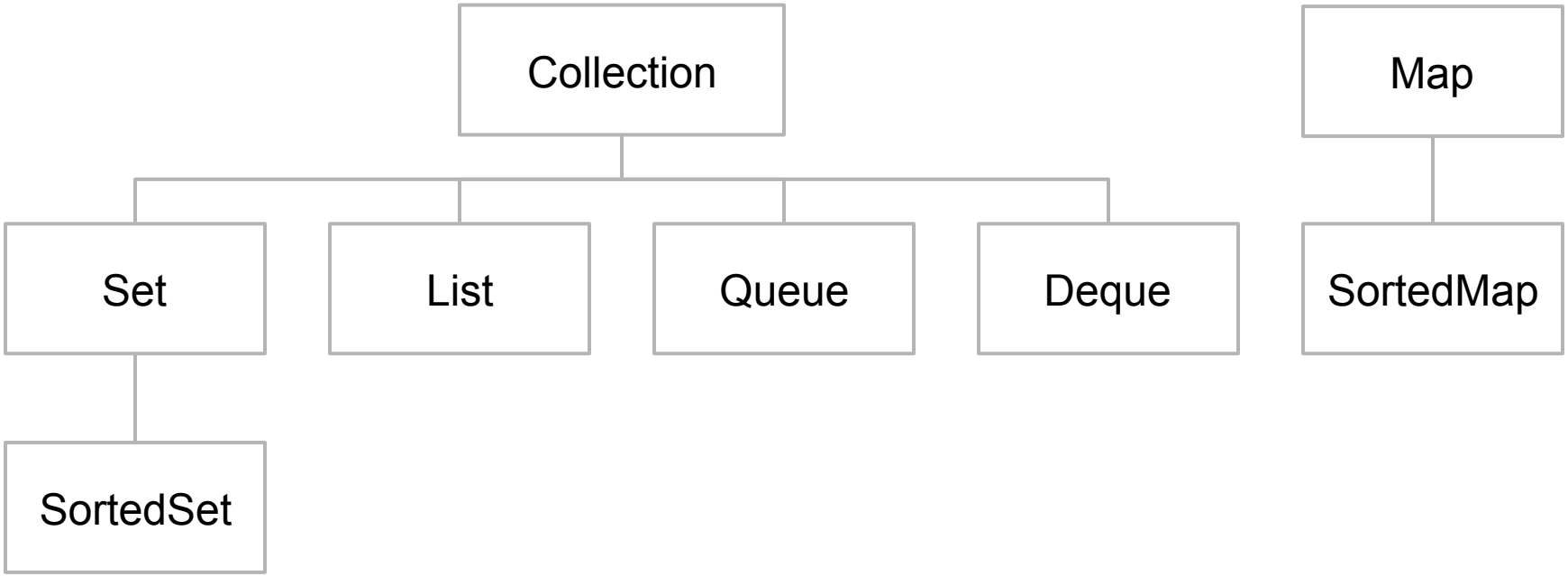
https://docs.oracle.com/javase/8/docs/api/java/util/AbstractList.html

Unsupported Operations

- Collections class has methods to create *unmodifiable* versions of each collection type
  - Throws UnsupportedOperationException to prevent modification operation
- Same exception thrown by implementations that do not support *optional* methods in hierarchy

https://docs.oracle.com/javase/8/docs/api/java/lang/UnsupportedOperationException.html

Core Interface Hierarchy



http://docs.oracle.com/javase/tutorial/collections/interfaces/index.html

Abstract Classes

- Implement interfaces in Collection hierarchy and provide basic implementations where possible
- Includes AbstractCollection, AbstractMap, AbstractList, AbstractSequentialList, AbstractSet, and AbstractQueue
- Usually what is extended by actual implementations

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science